
Sissejuhatus infotehnoloogiasse

Loengu ülevaade. Interneti rakendused.

Meeldetuletus: baasprotokollid, HTTP protokoll

Failide tõmbamine ja serveerimine:

- Kuidas http abil faili tõmmata

- Kuidas http abil faili serveerida

- Info saatmine serverile: cgi protokoll

Mis brauseril sees on:

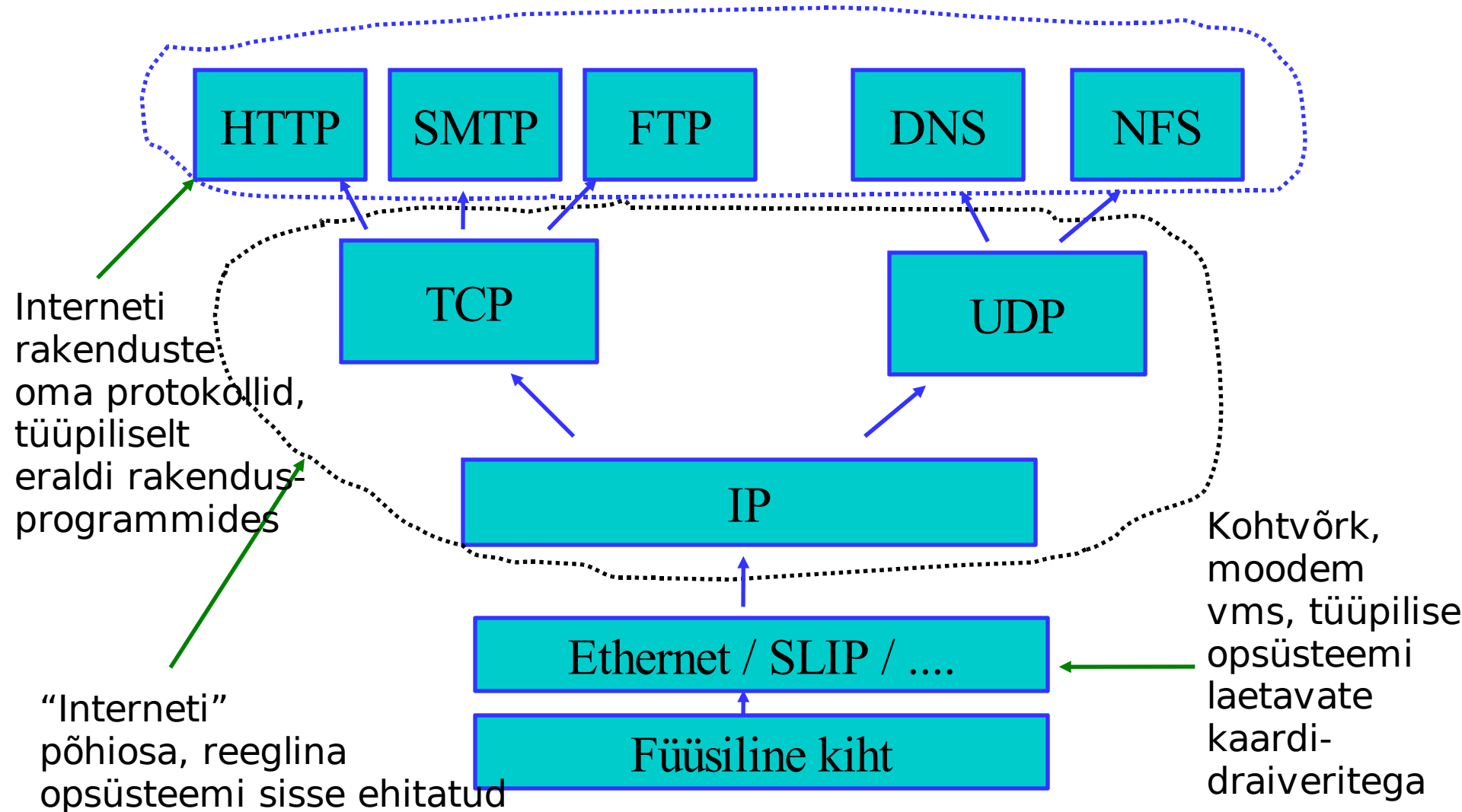
- Html, css, javascript jne

- Xml ja xslt

XML

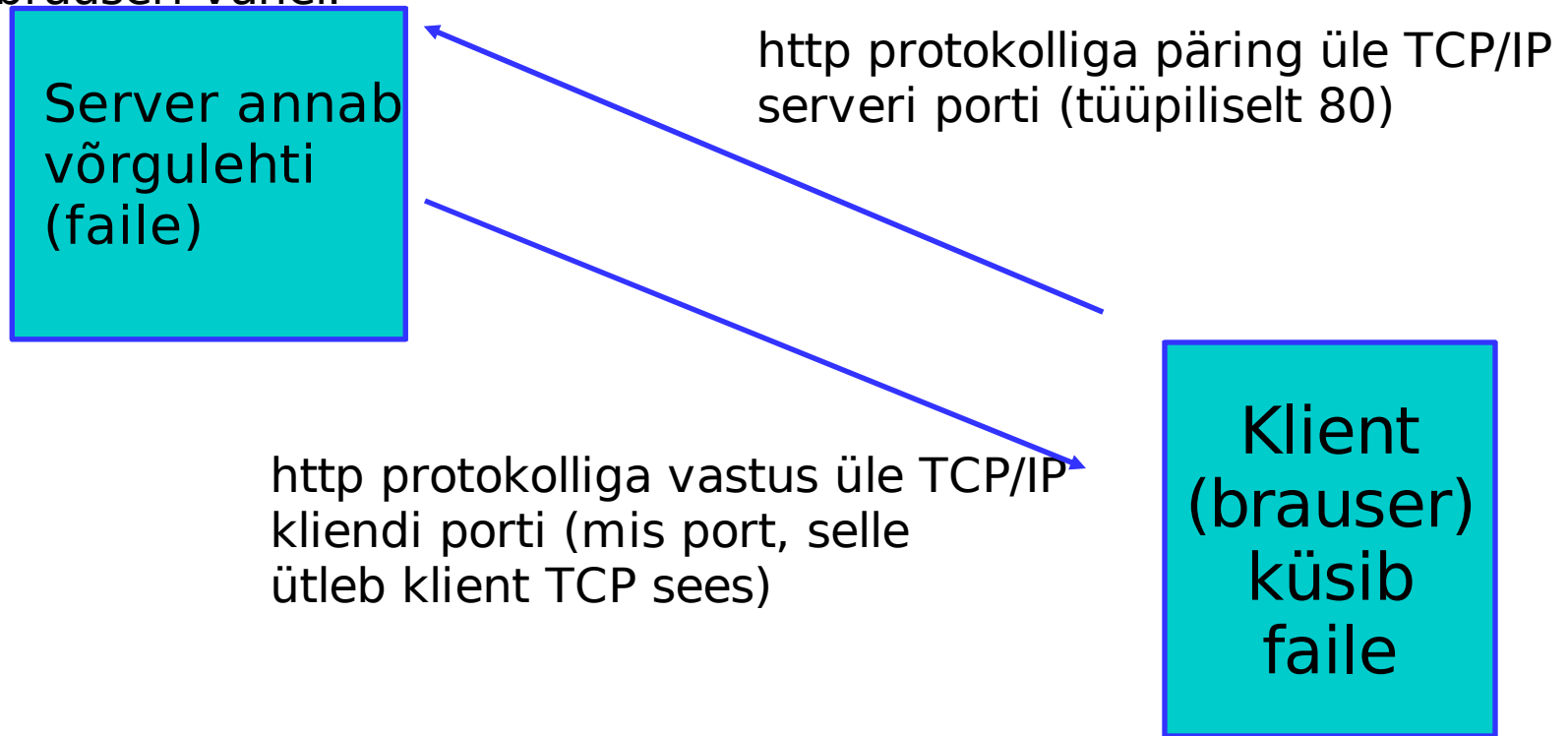
Andmete esitus, rdf ja ontoloogiad

Kokkuvõte protokollindusest internetis



HTTP ühendused: failide küsimine ja nende andmine

HTTP on omaette protokoll, mida kasutatakse veebilehtede, piltide, tekstifailide, zip failide jne jne saatmiseks veebiserveri ja brauseri vahel.



Kuidas html-faili tõmmata?

Vaatame lihtsat java-koodi näidet:

<http://java.sun.com/docs/books/tutorial/networking/urls/readingURL.html>

Lihtsamad brauserid ja brauseritaolised rakendused:

lynx

wget

curl

....

Veebilehtede (failide) serveerimine

Veebilehed on lihtsalt failid, mille ette server paneb http-päise

Veebileht (fail) võetakse reeglina kas:

- olemasoleva failina arvuti kettal

- tekstina, mille teeb iga kord uuesti mõni programm (cgi)

- Programm võib olla külge-ehitatud veebiserverile (php, mod-perl jne)

- ... või olla eraldi programm, mille server käivitab (klassikalised cgi-programmid)

Vaatame tiny serverit näitena:

http://www.lambda.ee/images/8/89/Itsissejuhatus_tiny.zip

Andmete saatmine serverile: cgi

URL-i lõpus (peale ?) oleva teksti saab serveri kutsutav programm (nn cgi-programm) kätte

Brauser saadab vormi info cgi-kodeeritult nii:
a=1&b=35 jne, kus a ja b jne on välja nimed, võrduse järel aga kasutaja sisestatud väärtused

Vaatame cgi näiteid tiny serveri pealt:

`http://localhost:8000/cgi-bin/proov.sh?asas`

`http://localhost:8000/cgi-bin/adder?2&3`

`http://localhost:8000/cgi-bin/proov.py?a=2&b=14`

Brauser oskab mitut keelt

Iga normaalne brauser mõistab:

Html

Css

Javascript

Xslt

....

Kust lugeda:

Ametlikud standardid: <http://www.w3.org>

Tutorialid: <http://www.w3schools.com>

HTML:

Teksti paigutamise / lehe kujundamise keel

Näited kohapeal

Loe: <http://www.w3schools.com/html/default.asp>

Standard: <http://www.w3.org/TR/html401/>

Mis on XHTML:

Praktiliselt seesama, mis HTML (samad tagid jne)

Tagid peavad olema väikeste tähtedega

Peab olema korrektses XML süntaksis (tagi suletud jne)

Paar XML-i lisaknihvi ka (namespaced jne)

Eriti täpset teksti paigutust ja kujundust võimaldav keel HTML täienduseks

Näited kohapeal

Uuri lisaks: <http://www.w3schools.com/css/default.asp>

**Brauseri programmeerimiskeel: javascripti
programmid töötavad otse brauseris: muudavad
htmli, css-i, võtavad ühendust serveriga jne jne**

Näited kohapeal

Uuri lisaks: <http://www.w3schools.com/js/default.asp>

Javascript, CSS, asynchronous queries

DHTML tähistas: HTML+CSS+Javascript

AJAX tähistab: HTML+CSS+Javascript+async. Queries

Vaatame näidet:

<http://www.papermountain.org/demos/live/#>

eXtensible Markup Language

XML on:

- Struktureeritud teksti esitamise formaat.

- XML standard ütleb, kuidas teksti struktuuri märgistada.

- Üllatavalt lihtne.

- Väga levinud ja ei ilmuta kadumise märke: vastupidi.

XML ei ole:

- Programmeerimiskeel.

- Mida intelligentset või automaatset.

- Imevahend.

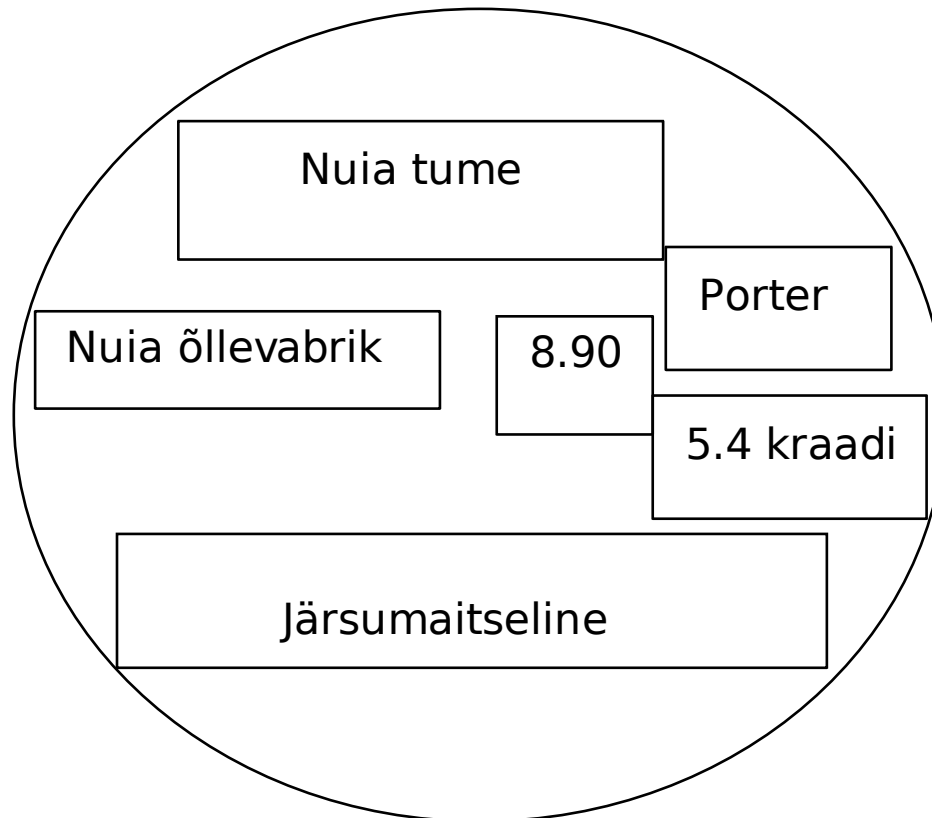
- Kasutu.

Tüüprakendus

Meil on mingid andmed.

Neid on vaja faili kirja panna ja hiljem arvutiga töödelda ja teistele edasi saata.

Olgu meil info mingi õllesordi kohta:



Kuidas seda faili kirja panna?

Variant 1: inimkeelsena.

Nuia tumeda tootja on Nuia õllevabrik. Porterit tüüpi õlu. Kangus 5.4 kraadi. Järsumaitseline. Hind 8.90.

Oleks hea variant, aga arvutiga praktiliselt lootusetu töödelda.

Variant 2: CSV (comma separated values)

“Nuia tume”, “Nuia õllevabrik”, Porter, 8.90, 5.4, Järsumaitseline

Päris hea ja väga levinud variant.

Failis ei saa öelda, mis tähestikus (õ, ä jne kodeering).

Vaja on eraldi öelda, mida väljad tähendavad (selleks saab kasutada päiserida).

Põhiprobleem: järgmisel slaidil

CSV ja muude tabelkujude põhiprobleem

CSV ja muud tabeliformaadid esitavad infot tabelina:

See ei sobi hästi, kui tabeli ruutudes peaks olema alamtabelid.

Näiteks, kui:

Tahaksime sisestada aadressi: riik, linn/küla, tänav, maja/krt

Hinna juures on hinnaühik (EEK, EUR, jne)

Jne

SQL andmebaasides kasutatakse siis abitabeleid ja neid siduvaid kunstlikke ID-numbreid, mis pole seotud algse dataga.

XML viis eelmist infot kirja panna

XML-s “pakitakse” infoväljad kirjeldavate nn tag-de vahele:

```
<olu>
  <nimi>Nuia tume</nimi>
  <tootja>Nuia õllevabrik</tootja>
  <tyyp>Porter</tyyp>
  <hind>8.90</tyyp>
  <kangus>5.4</kangus>
  <maitse>Järsumaitseline</maitse>
</olu>
```

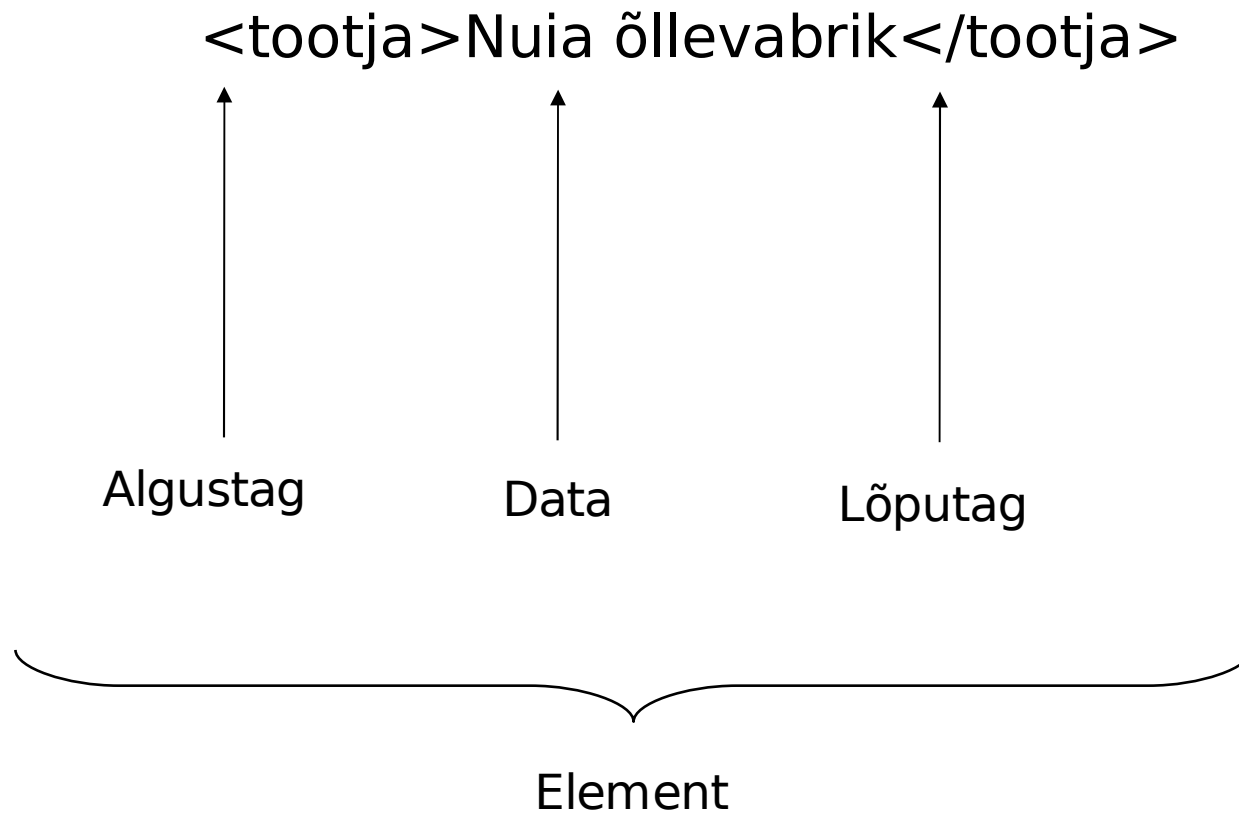
Pluss: saab töödelda arvutiga nagu CSV-d

Pluss: inimesele kohmakas lugeda, kuid siiski üheselt arusaadav.

Miinus: tekst on pikem, kui CSV puhul.

Sellest saab üle: kokkusurumine (zip, gz, jms pakkijad).

XML tagid



Tagide põhieelis

Elemendid võivad sisaldada teisi elemente:

<olu>

 <nimi>Nuia tume</nimi>

 <tootja>

 <nimi>Nuia õllevabrik</nimi>

 <aadress>

 <küla>Nuia</küla>

 <tänav>Tallinna mnt</tänav>

 <maja>2</maja>

 </aadress>

 </tootja>

 <tyyp>Porter</tyyp>

 <hind>8.90</tyyp>

 <kangus>5.4</kangus>

 <maitse>Järsumaitseline</maitse>

</olu/>

HTML ja XML

Info pannakse “tag”-ide vahele: infol on sildid

HTML: “tag”-del visuaalne semantika

Siin on paks tekst

<i>Siin on kaldkirjas tekst</i>

XML: “tag”-del semantika puudub

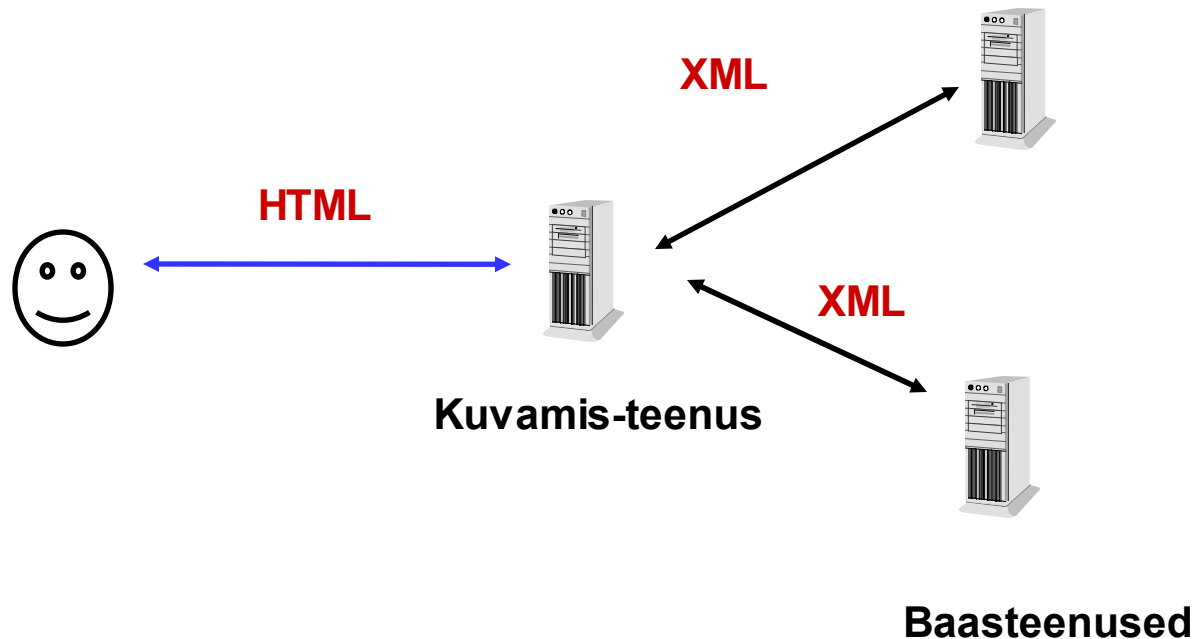
<autor>Tanel Tammet</autor>

<minuaadress>Kuiv 9</minuaadress>

Võrguteenused

Probleem: väga raske, pea võimatu on teha programmi, mis loeks teisest serverist HTML-lehekülgi.

Idee: teeme võrgulehekülgi XML-is, nii et programm teises arvutis suudaks neid lugeda



Protseduraalne vs deklaratiivne

Protseduraalne info esitus:

info on programmi sisse kodeeritud.

Deklaratiivne info esitus:

info on eraldi reeglitena (failis), mida reegleid kasutav algoritm rakendab

Mis puudub?

Üks programm peab teise programmi antud XML-ist
SISULISELT aru saama

Õllepartiide otsimise server:

Hulgimüüja A annab infot:

```
<beer>  
  <name>Guinness</name>  
  <price>100</price>  
</beer>
```

Hulgimüüja b annab infot:

```
<olu>  
  <mark>Guinness</mark>  
  <hind>100</hind>  
</olu>
```

Hulgimüüja C annab infot:

```
<porter>  
  <name>Guinness</name>  
  <price>100</price>  
</porter>
```

Erinevad keeled

Peame teadma, et:

BEER = ÕLU
PORTER on ka BEER

Ei ole lootust teha “ette valmis” kõiki keeli

On olemas alamkeeled (üldine toodete keel, õllede keel Eestis, Tshehhis, Hiinas, õllede keel Tartu Õllefabrikus ja Sakus, ...)

Keeli on vaja omavahel tõlkida

Tõlkimiseks on vaja reegleid ja elementaarset arusaamist

Semantic web: projekt

Eesmärk: pakkuda XML-põhiseid keeli ja luua standardeid, mille abil kirjeldada asju (internetis) ja (XML) keeltevahelisi seoseid.

Soovitav tulemus: saab teha tarkvara, mille abil XML-datat saab ühest programmist teise saata, nii et info kaduma ei lähe.

Esmajoones vaja:

- objektide kirjeldamise keel
- mõistete omavaheliste seoste ja reeglite keel

Projekti juht: Tim Berners-Lee (<http>, www ja esimese brauseri autor, w3c juht).

Ülevalt alla:

Tõestuste keel

Täisloogika

Ontoloogiad

Ontoloogiakeel (Owl ja tema alamhulgad)

Asjade kirjelduskeel (RDF ja RDFS)

Nimede unikaliseerimine (XML namespaces)

Süntaks (XML)

Transport (http, https, SOAP, tcp/ip ...)

Nimede unikaliseerimine: XML namespaces

Idee:

iga tagi ette saab kirjutada, mis keeles ta on <keel:tag>
keelte nimedeks kasutatakse URI-sid (URL-e)

Näide:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:ex="http://www.example.org/terms/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <ex:creation-date>August 16, 1999</ex:creation-date>
    <ex:language>English</ex:language>
    <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>
</rdf:RDF>
```

Asjade kirjeldamine: RDF

Idee:

Kõik kirjeldused on kolmikud: [omadus, ressurss, objekt]

Predikaadina: omadus(ressurss, objekt)

Kõik need kolm võivad olla URI-d

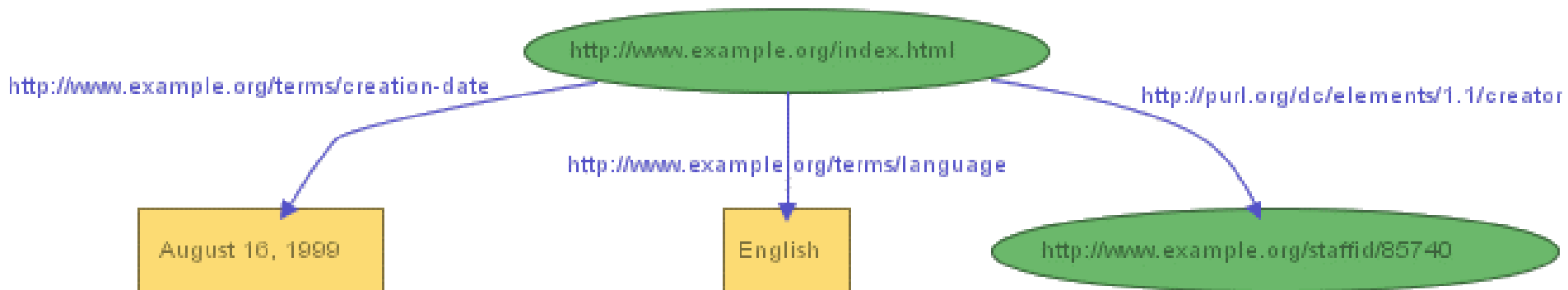
Kirjelduskeel RDF

<http://www.example.org/index.html> has a creation-date whose value is August 16, 1999

<http://www.example.org/index.html> has a language whose value is English

<http://www.example.org/index.html> has a creator indicated by

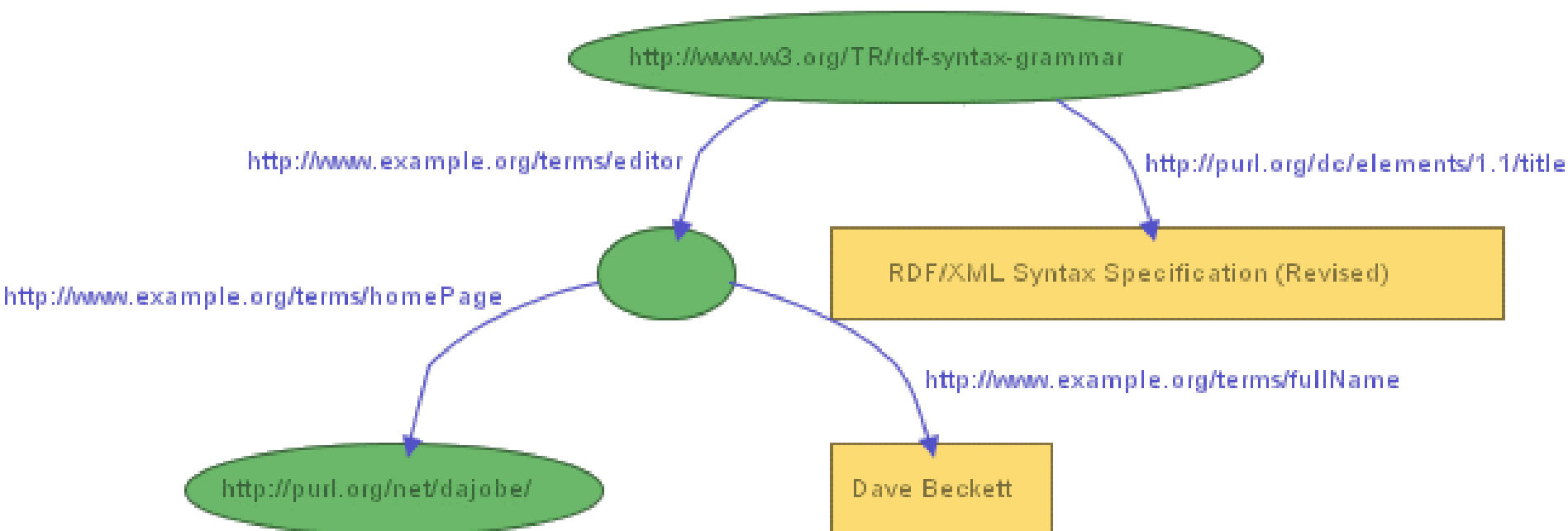
<http://www.example.org/staffid/85740>



kirjelduskeel RDF

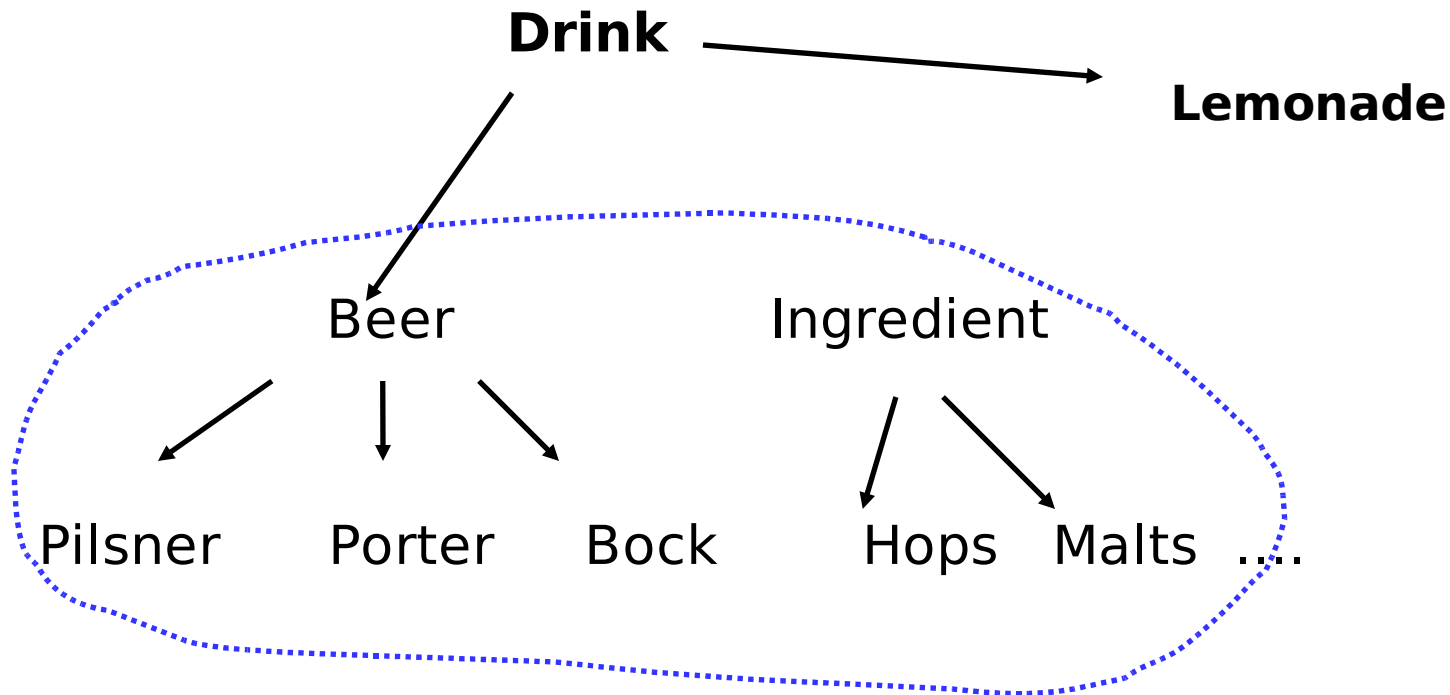
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:ex="http://www.example.org/terms/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <ex:creation-date>August 16, 1999</ex:creation-date>
    <ex:language>English</ex:language>
    <dc:creator
      rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>
</rdf:RDF>
```

kirjelduskeel RDF: sügavam puu



Ontoloogiad

Mis on ontoloogiad: “maailma mudelid” ehk mõistete hierarhiad



Eri valdkondades on oma ontoloogiad

Ilmselt ei õnnestu panna kõiki programme kasutama samu “standardontoloogiaid”